

# ► Compte rendu du projet Calculatrice.

Création d'une calculatrice simple en HTML, CSS et JavaScript

Ismail Mabrouki ► ESIP ► 5/3/2023

Le code présent est une calculatrice simple codée en HTML, CSS et JavaScript. La partie CSS est responsable de la mise en page de la calculatrice, tandis que la partie JavaScript gère les fonctionnalités de la calculatrice.

Le CSS définit une disposition de la calculatrice à l'aide de flexbox et d'une grille. La calculatrice a une largeur de 200 pixels et une bordure de 1 pixel solide noire. Les boutons de la calculatrice ont un remplissage de 10 pixels, une taille de police de 16 pixels et une bordure arrondie de 5 pixels.

Le JavaScript définit quatre fonctions pour la calculatrice :

- `addNumber(num)` : cette fonction ajoute le nombre correspondant au bouton cliqué à l'affichage.
- `addOperator(op)` : cette fonction ajoute l'opérateur correspondant au bouton cliqué à l'affichage.
- `calculate()` : cette fonction calcule le résultat de l'expression affichée à l'aide de la fonction `eval()` et affiche le résultat.
- `clearDisplay()` : cette fonction efface l'affichage de la calculatrice.
- 

En résumé, ce code est un exemple de calculatrice fonctionnelle que l'on peut facilement intégrer à un site web en utilisant les langages HTML, CSS et JavaScript

# Compte rendu du projet Calculatrice.

Création d'une calculatrice simple en  
HTML, CSS et JavaScript

## Le code HTML de base

- La page utilise le DOCTYPE HTML5
- Elle contient une balise `<html>` qui définit le début du document
- Dans la balise `<head>`, on trouve le titre de la page et un lien vers une feuille de style externe
- Dans la balise `<body>`, on trouve la structure de base de la calculatrice, qui est composée de deux divs imbriquées.

## La première div

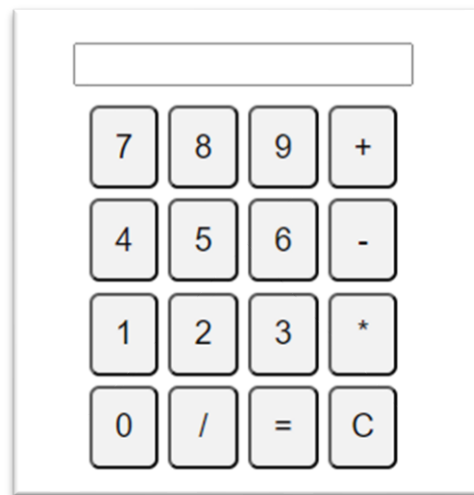
- La première div a la classe "container"
- Elle contient une autre div avec la classe "calculator"

## La deuxième div

- La deuxième div a la classe "calculator"
- Elle contient deux autres divs : une pour l'affichage du résultat et l'autre pour les boutons de la calculatrice

## L'affichage du résultat

- L'affichage du résultat est une input de type "text" avec l'ID "result"
- L'attribut "readonly" empêche l'utilisateur de modifier le contenu de l'input



## Les boutons

- Les boutons sont créés avec la balise `<button>`
- Chaque bouton a un attribut "onclick" qui appelle une fonction JavaScript avec un argument correspondant au chiffre ou à l'opérateur arithmétique
- Les boutons pour les nombres ont du texte à l'intérieur de la balise
- Les boutons pour les opérateurs ont leur symbole à l'intérieur de la balise

## Le fichier JavaScript externe

- La balise `<script>` dans la dernière ligne de la page web charge le fichier "app.js"
- Ce fichier contient les fonctions JavaScript qui sont appelées par les boutons de la calculatrice

## Explication du CSS pour le conteneur

- Le code CSS présenté correspond à la mise en forme du conteneur principal du projet Calculatrice.
- La première instruction définit la largeur du conteneur à 100% de la largeur de l'écran, tandis que la hauteur est définie à 100vh (100% de la hauteur de la fenêtre).
- Ensuite, la propriété `display` est utilisée avec la valeur "flex" pour que les éléments du conteneur soient organisés en ligne, avec un centrage horizontal et vertical grâce à "justify-content" et "align-items".
- Enfin, "text-align" est utilisé pour centrer le texte à l'intérieur du conteneur.

```
.container {  
  width: 100%;  
  height: 100vh;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  text-align: center;  
}
```

```
.calculator {  
  width: 200px;  
  border: 1px solid black;  
  padding: 10px;  
  display: flex;  
  flex-direction: column;  
  justify-content: center;  
  align-items: center;  
  height: fit-content;  
}
```

## Classe :calculator

- La classe `.calculator` est utilisée pour définir les styles pour notre calculatrice
- Les propriétés suivantes sont définies :
  - `width: 200px` : permet de définir la largeur de la calculatrice sur 200 pixels
  - `border: 1px solid black` : permet de définir une bordure solide de 1 pixel de large et de couleur noire
  - `padding: 10px` : permet de définir une marge

intérieure de 10 pixels  
autour de la calculatrice

- display: flex : permet d'utiliser la disposition flexible pour organiser les éléments
- flex-direction: column : permet de disposer les éléments enfants en colonne
- justify-content: center : permet de centrer horizontalement les éléments enfants
- align-items: center : permet de centrer verticalement les éléments enfants
- height: fit-content : permet de définir la hauteur de la calculatrice en fonction de son contenu

#### display

- La classe .display est utilisée pour définir les styles pour l'affichage de la calculatrice
- La propriété margin-bottom: 10px permet de définir une marge inférieure de 10 pixels pour séparer l'affichage des boutons

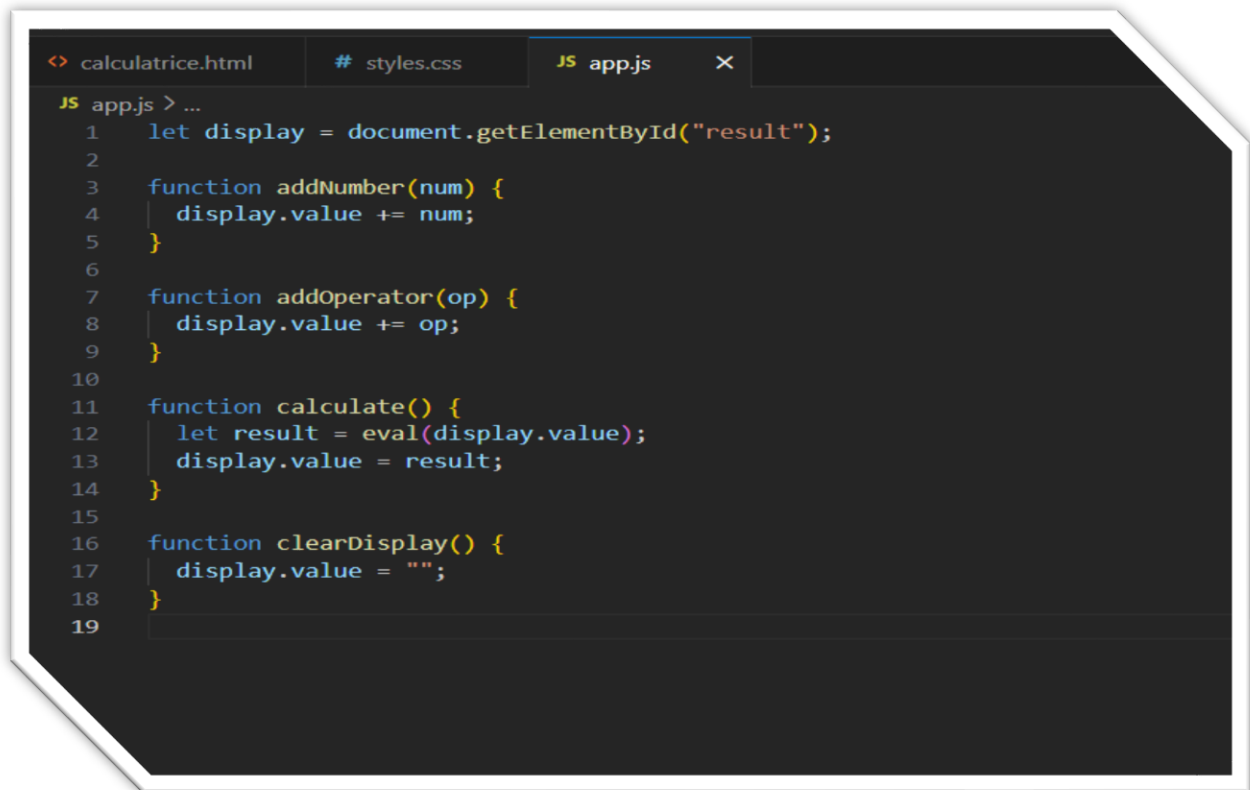
#### buttons

- La classe .buttons est utilisée pour définir les styles pour les boutons de la calculatrice
- Les propriétés suivantes sont définies :

- display: grid : permet d'utiliser la disposition en grille pour organiser les boutons
- grid-template-columns: repeat(4, 1fr) : permet de définir 4 colonnes de largeur égale avec une fraction de l'espace disponible
- grid-gap: 5px : permet de définir un espace de 5 pixels entre chaque bouton

#### button

- La classe button est utilisée pour définir les styles pour chaque bouton de la calculatrice
- Les propriétés suivantes sont définies :
  - padding: 10px : permet de définir une marge intérieure de 10 pixels pour les boutons
  - font-size: 16px : permet de définir la taille de police à 16 pixels
  - border-radius: 5px : permet de définir un rayon de bordure arrondi de 5 pixels pour les boutons.

A screenshot of a code editor with three tabs: 'calculatrice.html', '# styles.css', and 'JS app.js'. The 'JS app.js' tab is active, showing the following JavaScript code:

```
JS app.js > ...
1  let display = document.getElementById("result");
2
3  function addNumber(num) {
4    display.value += num;
5  }
6
7  function addOperator(op) {
8    display.value += op;
9  }
10
11 function calculate() {
12   let result = eval(display.value);
13   display.value = result;
14 }
15
16 function clearDisplay() {
17   display.value = "";
18 }
19
```

D'après le code fourni, nous pouvons conclure que la calculatrice a été réalisée avec succès en utilisant HTML, CSS et JavaScript. Le code HTML a été utilisé pour créer la structure de base de la calculatrice, le CSS pour la mise en forme visuelle et le JavaScript pour la fonctionnalité de la calculatrice. Le JavaScript utilise des fonctions pour ajouter des nombres, des opérateurs, effectuer le calcul et effacer l'écran. Le code est facile à comprendre et à modifier, ce qui le rend adapté pour les projets de développement web similaires. La calculatrice est fonctionnelle et prête à l'emploi pour effectuer des calculs simples.